# 通過テスト１１　解答

## 11 例外

第１刷では問題１に訂正があります
第２刷以降は修正されています

1.　A, E

＜問題の訂正＞ return が欠けています．return を追加して考えてください．
（return がないとコンパイルエラーになります）

```
int doIt(String s){
    try{
        return s.equals("abc")? 0:1;
    }
    ・・・・・
}
```

【解説】実行時例外が起こるのは equal メソッドを使っている所で，s が null だった場合．これは実行時例外なので NullPointerException がそのスーパークラスである RuntimeException で補足できる．C,F,G,H の実行時例外では NullPointerException を補足できない．また，B,D はチェック例外であり，指定するとコンパイルエラーになる．正解は A と E．

2.　B

【解説】次々にメソッドが呼び出され，最後に呼び出される sizeC メソッドで文字列を数値に変換しようとするので例外（NumberFormatExceptiong）が発生する．呼び出し元の SizeB では NullPointerException しか補足しないので，さらに sizeA メソッドに戻り，ここで補足されて“ア”が出力される．例外は sizeA メソッドで補足されるので main メソッドには波及しない．

3.　F

【解説】yellow メソッドが例外 NumberFormatException を投げるが，呼び出し元の red メソッドはそれを補足する catch ブロックがない．しかし，Finally ブロックがあるのでまず"red"と表示してから，呼び出し元の main メソッドへ戻る．ただ，main でも例外を補足していないので，"main"と表示する前に，例外が JVM に波及してプログラムは停止する．正解は F．

4.　C

【解説】Exec クラスの disp メソッドは Master クラスの disp メソッドをオーバーライドしている．オーバーライドではスーパークラスより狭い範囲の例外を投げてよいが，例外を投げるメソッドでは throws 宣言が必要である．問題では Exec クラスの disp メソッドに throws IOException という宣言がないのでコンパイルエラーになる．